College of Professional and Continuing Education (CPCE), an affiliate of PolyU

THE HONG KONG POLYTECHNIC UNIVERSITY
HONG KONG COMMUNITY COLLEGE

Subject Title	: Data Structures	**Subject Code**	: CCN2239
Session	: Semester One, 2015/16	**Time**	: 09:30 – 12:30
Date	: 17 December 2015	**Time Allowed**	: 3 Hours
Subject Examiner(s)	: Dr Pat CHAN Dr Joseph SO		

This question paper has a total of **TWENTY-TWO** pages (including this covering page).

Instructions to Candidates:

1. There are THREE sections in this paper.
   - Section A (40%) – Multiple-choice Questions. Answer ALL questions in this section on the multiple-choice answer sheet provided. Each question carries 1 mark.
   - Section B (40%) – Short Questions. Answer any FOUR of the FIVE questions in this section in the answer book provided. Each question carries 10 marks.
   - Section C (20%) – Long Questions. Answer any ONE of the TWO questions in this section in the answer book provided. Each question carries 20 marks.
2. Candidates are NOT allowed to retain the multiple-choice answer sheet, the answer book and the examination question paper.
3. Show all your work clearly and neatly. Marks will be deducted for untidy work.
4. Reasonable steps should be shown.
5. All programming code must be written in Java programming language.

Authorised Materials:

	YES	NO
CALCULATOR	[ ]	[✓]
SPECIFICALLY PERMITTED ITEMS	[ ]	[✓]

DO NOT TURN OVER THE PAGE UNTIL YOU ARE TOLD TO DO SO

College of Professional and Continuing Education (CPCE), an affiliate of PolyU

Section B (40%) – Short Questions

Answer any FOUR of the FIVE questions in this section in the answer book provided. Each question carries 10 marks. If more than FOUR questions are answered, only the first FOUR questions answered will be marked.

Question B1

Suppose we have the following nodes with (key, element) pair:

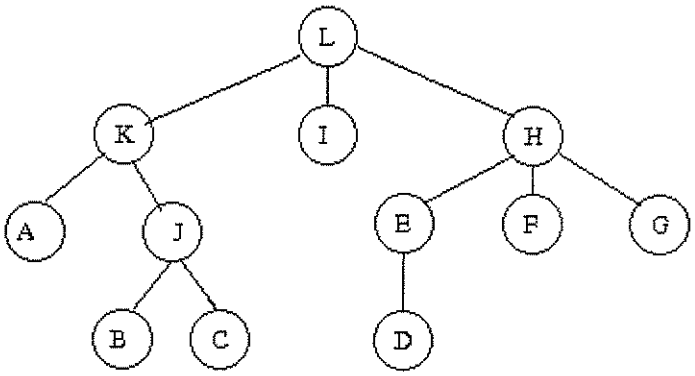
(2, A), (5, B), (1, C), (7, D), (4, E), (6, F), (3, G)

(a) Draw the max-heap (requires that key at a node is greater than the key of its two children) associated with inserting the keys in the order A, B, G, F, C, E, D. (6 marks)

(b) What is the maximum number of nodes for a max-heap with height of 4? What is the minimum number? Explain your answer. (4 marks)

Question B2

(a) Given the following tree:



(i) List the sequence of nodes visited by preorder traversal. (2 marks)

(ii) List the sequence of nodes visited by postorder traversal. (2 marks)

(iii) List the sequence of nodes visited by level order traversal. (2 marks)

(b) Write the pseudocode of a level order traversal for a *binary tree*. (4 marks)

College of Professional and Continuing Education (CPCE), an affiliate of PolyU

Question B3

You are given the following java program.

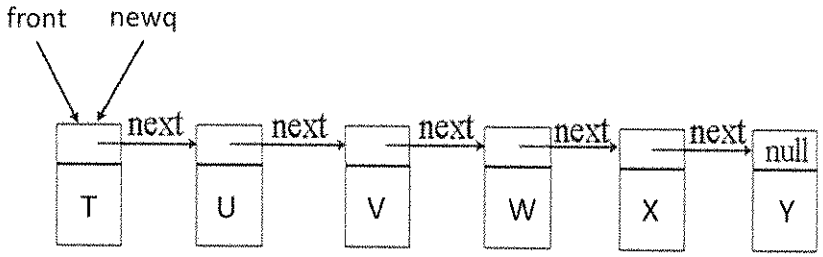
```
public class Printing {

	public int upper(int m) {
		int u = 0;
		for (int j = 0; j < m; j++) {

			for (int i = 0; i <= j * 2; i++) {
				System.out.print("*");
				u += i;
			}
			System.out.println();
		}
		return u;
	}

	public int lower(int m) {
		int v = 1;
		for (int j = 1; j < m; j++) {

			for (int i = 0; i < j; i++) {
				System.out.print("^");
			}
			System.out.println(v);
			for (int i = (m - 1) * 2; i >= j * 2; i--) {
				System.out.print("*");
			}
			v *= j;
			System.out.println();
		}

		return v;
	}

	public int work(int m) {
		if (m <= 0)
			return 1;
		System.out.println(m);
		return work(m - 3) + work(m - 2);

	}

	public static void main(String[] args) {
		Printing p = new Printing();
		System.out.println(p.upper(5));
		System.out.println();
		System.out.println(p.lower(6));
		System.out.println();
		System.out.println(p.work(7));

	}
}
```

Show the output when successfully executing `Printing` in java. (10 marks)

College of Professional and Continuing Education (CPCE), an affiliate of PolyU

Question B4

The following figure shows the linked list structure in which the pointers `front` and `newq` point at the first node and the pointer `next` points at the next node.



(a) After executing the following Java statements,

```
newq = front.next.next.next.next;
front.next.next.next = front.next;
front.next = front.next.next;
front.next.next.next = null;
```

the linked list in the above will be changed. Draw all original nodes with the new links.

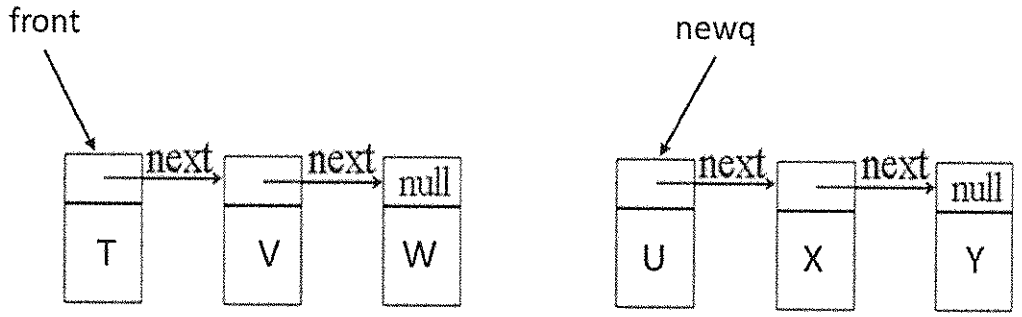
(4 marks)

(b) What will be happened when a node has no pointer referencing on it? (1 mark)

(c) For the original linked list **before** part (a), complete the following Java statements such that after executing these statements, the linked list in the above will change to the two lists in figure below and the pointer `front` and `newq` will point one list each.

(Note: modifying any of the following given parts of the Java statements or adding Java statements is not allowed.)

```
newq = _____________________________
front_____________ = _____________________
newq_____________ = _____________________
front_____________ = _____________________
```



(5 marks)

College of Professional and Continuing Education (CPCE), an affiliate of PolyU

Question B5

(a) Choose the **BEST** sorting algorithm for each of the following situations.

(i) I am sorting data that is stored over a network connection. Based on the properties of that connection, it is extremely expensive to "swap" two elements. But looping over the elements and looking at their values is very inexpensive. I want to minimize swaps above all other factors.

(ii) I have a fast computer with many processors and lots of memory. I want to choose a sorting algorithm that is fast and can also be parallelized easily to use all of the processors to help sort the data.

(iii) I have an array that is already sorted. Periodically, some new data comes in and is added to the array at random indexes, messing up the ordering. I need to re-sort the array to get it back to being fully ordered. I do not want to use very much additional memory during the sort.

(6 marks)

(b) Briefly describe how the merge-sort algorithm works. (4 marks)

- End of Section B -

Section C (20%) – Long Questions

Answer any ONE of the TWO questions in this section in the answer book provided. Each question carries 20 marks. If more than ONE question is answered, only the first ONE question answered will be marked.

Question C1

Given the following Java code for sorting

```
public class SortingM1 {

      static Comparable[] a;

      public static void goodSort(Comparable[] a, int leftEnd, int rightEnd) {
            if (leftEnd >= rightEnd)
                  return;

            for (int i = 0; i< a.length; i++)
                  System.out.print(a[i] + " ");
            System.out.println();

            Comparable pivot = a[leftEnd];
            int i = leftEnd,
            j = rightEnd;

            while (i < j) {
                  while (j > i && a[--j].compareTo(pivot) >= 0)
                        ;
                  if (j > i)
                        a[i] = a[j];
                  while (i < j && a[++i].compareTo(pivot) <= 0)
                        ;
                  if (i < j)
                        a[j] = a[i];
            }
            a[j] = pivot;
            goodSort(a, leftEnd, j);
            goodSort(a, j + 1, rightEnd);
      }
```

College of Professional and Continuing Education (CPCE), an affiliate of PolyU

Question C1 (continued)

```
    public static void main(String[] args) {
        Integer[] a = { new Integer(10), new Integer(9), new Integer(14),
                        new Integer(12), new Integer(3), new Integer(19),
                        new Integer(8), new Integer(7), new Integer(8) };

        System.out.println("The elements are");
        for (int I = 0; I < a.length; i++)
            System.out.print(a[i] + " ");
        System.out.println();

        goodSort(a, 0, a.length);

        System.out.println("The sorted order is");
        for (int I = 0; I < a.length; i++)
            System.out.print(a[i] + " ");
        System.out.println();
    }
}
```

(a) What is the sorting algorithm used in `goodSort`? (1 mark)

(b) Show the output when successfully executing `SortingM1` in java. (5 marks)

(c) You are further given the following java program with a `FruitList` class. Write a single line code to replace the `<missing statement>` so that it can sort the fruits by using the method `goodSort`. (2 marks)

```
import java.util.Arrays;

public class FruitList {

   public static void main(String[] args) {
          Fruit[] fruits = new Fruit[5];

          Fruit apple = new Fruit("Apple", "Red rounded", 20);
          Fruit pineappale = new Fruit("Pineapple", "Heavy and yellow",
          100, 60);
          Fruit orange = new Fruit("Orange", "Orange rounded", 80);
          Fruit banana = new Fruit("Banana", "Soft and yellow", 90, 200);
          Fruit pear = new Fruit("Pear", "Yellow and crispy", 20);

          fruits[0] = pineappale;
          fruits[1] = apple;
          fruits[2] = banana;
          fruits[3] = orange;
          fruits[4] = pear;

          <missing statement>  //code to be completed in part (c)

          for (int i = 0; i < fruits.length; i++) {
              System.out.println(fruits[i].getFruitName() + " Quantity : "
                          + fruits[i].getQuantity());
          }
          System.out.println("There are total number " + Fruit.getTotal()
                    + " of fruits");
          System.out.println("The expected total weight is  "
                    + Fruit.getTotalWeight(fruits) + " of fruits");
   }
}
```

College of Professional and Continuing Education (CPCE), an affiliate of PolyU

It will give the following results after successful execution:

```
Pineapple Apple Banana Orange Pear
Pear Apple Banana Orange Pineapple
Pear Apple Banana Orange Pineapple
Pear Apple Banana Orange Pineapple
Pear Apple Orange Banana Pineapple
Pear Quantity : 20
Apple Quantity : 20
Orange Quantity : 80
Banana Quantity : 90
Pineapple Quantity : 100
There are total number 310 of fruits
The expected total weight is  24120.0 of fruits
```

(d) For the execution of the programs given in part (c), complete the specified methods (i) to (v) for the `Fruit` class shown below.

(i) Add a constructor method to allow all object instances of `Fruit` successfully initialized in `FruitList`. (3 marks)

(ii) Add a method `getQuantity` that returns the quantity of a specific fruit. (2 marks)

(iii) Add a method `getTotal` that returns the total number of all fruits. (2 marks)

(iv) Add a method `getTotalWeight` so that `Fruit.getTotalWeight(fruits)` in `FruitList` gets the total weight of all fruits. (3 marks)

(v) Add codes in `compareTo` in `Fruit` so that the object instance of `Fruit` can be sorted by the quantity. (2 marks)

Note that you may use the methods provided in the above code and you may show only the added Java code in your answer. Add the above static methods (i) to (v) to perform the required functions for the list.

```
public class Fruit implements Comparable<Fruit>{

   private String fruitName;
   private String fruitDesc;
   private int quantity;
   private double avgWeight;
   protected static int totalCount;

   public Fruit(String fruitName, String fruitDesc, int quantity) {

         this(fruitName, fruitDesc, quantity, 1.0);
   }
```

// code to be completed for part (d) (i)

```
   public String getFruitName() {
         return fruitName;
   }
```

College of Professional and Continuing Education (CPCE), an affiliate of PolyU

Question C1 (continued)

```
public void setFruitName(String fruitName) {
      this.fruitName = fruitName;
}
public String getFruitDesc() {
      return fruitDesc;
}
public void setFruitDesc(String fruitDesc) {
      this.fruitDesc = fruitDesc;
}
```

// code to be completed for part (d) (ii)

```
public void setQuantity(int quantity) {
      this.quantity = quantity;
}
```

// code to be completed for part (d) (iii)

```
public String toString() {
      return fruitName;
}
```

// code to be completed for part (d) (iv)

```
public int compareTo(Fruit compareFruit) {
```

// code to be completed for part (d) (v)

```
}
}
```

College of Professional and Continuing Education (CPCE), an affiliate of PolyU

Question C2

Suppose you have an `AVLNode` class that stores integers:

```
public class AVLNode {
      public int item;
      public AVLNode left;
      public AVLNode right;
      public AVLNode (int i, AVLNode l, AVLNode r){
            item = i;
            left = l;
            right = r;
      }
}
```

(a) Write a complete method that takes a height `h`, and returns a reference to the root of an AVL tree of height `h` that contains the minimum number of nodes. The integers stored in the nodes (the item instance variables) should satisfy the binary search tree property, and they should all be distinct (but they do not have to be consecutive). You can define helper methods and/or classes if you wish. (6 marks)

(b) Draw diagrams to show how to insert the following sequence of elements into an AVL tree, starting with an empty tree:

10, 20, 15, 25, 30, 16, 18, 19.

(4 marks)

(c) Draw diagrams to show how to delete the element 30 in the AVL tree that you got in part (b). (3 marks)

(d) What order should we insert the elements {1, 2, 3, 4, 5, 6, 7} into an empty AVL tree so that we do not have to perform any rotations on it? Explain your answer. (3 marks)

(e) Contrast the differences between AVL tree and binary search tree. (4 marks)

- End of Section C -

- END OF PAPER -